

REMARKS/ARGUMENTS

Claims 1-20 are pending. By this amendment, claims 1, 4, 9, 12, 15, and 18 are amended. New claims 21-24 are added. Claims 1, 4, 9, 12, 15, and 18 are amended to more precisely recite the features of the invention as implemented in a Java application, as suggested by the Examiner during the September 22, 2004 telephone conference. Support for the claim amendment can be found at least in original claims 1, 9, and 15. Support for the new claims can be found at least at page 3, line 21, page 3, lines 25-28, and page 4, lines 15-20 of the specification. No new matter is introduced. Reconsideration and prompt allowance of the claims is respectfully requested.

35 U.S.C. § 103 Rejections

Claims 1, 2, 3, 5, 6, 7, 9, 10, 11, 13, 15, 16, 17, and 19 are rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent 6,543,011 to Schumacher et al. (hereafter Schumacher) in view of U.S. Patent 6,636,246 to Gallo et al. (hereafter Gallo) and further in view of Microsoft Windows NT. This rejection is respectfully traversed.

Schumacher is directed to a method for recording events in Java. As acknowledged by the Examiner, Schumacher does not disclose or suggest attaching a custom glass pane to the window, wherein the mouse cursor is located in the window, and displaying a drag image approximate the mouse cursor using the custom glass pane. Gallo is directed to a method and system of providing a three dimensional spatial user interface (SUI) to a user of a computing device. Microsoft Windows NT shows a semi-transparent image of an object during a drag and drop operation in a Windows application. However, as discussed during the telephone conference, Gallo's system and Microsoft Windows NT cannot be applied to a Java application. Microsoft Windows system does not use Java language or Java library functions. Therefore, Schumacher, Gallo, and Windows NT, individually and in combination, do not disclose or suggest "attaching a custom glass pane to the window of the Java application, wherein the mouse cursor is located in the window; and displaying a drag image approximate the mouse cursor using the custom glass pane, wherein the drag image represents the dragging object and moves with the mouse cursor," as recited in amended claim 1 (emphasis added).

As noted in the Background section of the present application, "[h]owever, drag image support is not implemented in more recent versions of SUN JRE®. In other words, standard Java libraries don't support displaying an image of the object during dragging." The present application utilizes Java library functions to display a drag image during a drag and

drop operation in a Java application. Since the cited references do not disclose or suggest all of the features of amended claim 1, claim 1 is allowable.

Claims 2, 3, 5, 6, and 7 are allowable because they depend from allowable claim 1 and for the additional features they recite.

With respect to claim 9, for the same reason as discussed with respect to claim 1, Schumacher, Gallo, and Windows NT, individually and in combination, do not disclose or suggest “a custom glass pane attached to the window of the Java application, wherein the custom glass pane displays a drag image approximate the mouse cursor, and wherein the drag image represents the dragging object and moves with the mouse cursor,” as recited in amended claim 9 (emphasis added). Therefore, claim 9 is allowable.

Claims 10, 11, and 13 are allowable because they depend from allowable claim 9 and for the additional features they recite.

With respect to claim 15, for the same reason as discussed with respect to claim 1, Schumacher, Gallo, and Windows NT, individually and in combination, do not disclose or suggest “attaching a custom glass pane to the window of the Java application where the mouse cursor is located; and displaying a drag image approximate the mouse cursor using the custom glass pane, wherein the drag image represents the dragging object and moves with the mouse cursor,” as recited in amended claim 15 (emphasis added). Therefore, claim 15 is allowable.

Claims 16, 17, and 19 are allowable because they depend from allowable claim 15 and for the additional features they recite. Withdrawal of the rejection of claims 1, 2, 3, 5, 6, 7, 9, 10, 11, 13, 15, 16, 17, and 19 under 35 U.S.C. §103 (a) is respectfully requested.

Claims 4, 12, and 18 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Schumacher in view of Gallo and Microsoft Windows NT, and further in view of U.S. Patent 5,251,322 to Doyle et al. (hereafter Doyle). This rejection is respectfully traversed.

Doyle is directed to a method and apparatus for operating a computer graphics system to perform a conditional test on a node in a graphics data structure during the traversal of the graphics data structure by a structure walker. Similar to Gallo and Microsoft Windows NT, Doyle cannot be applied to a Java application. Therefore, Doyle does not cure the defect of Schumacher, Gallo, and Microsoft Windows NT and does not disclose or suggest all of the features of amended claims 1, 9, and 15. Claims 4, 12, and 18 are allowable because they depend from allowable claims 1, 9, and 15, respectively, and for the additional features they recite.

Furthermore, Doyle recites at column 50, line 54 “[a] client process 100 will call this routine to attach or detach a window (3DDC) to or from a graphics context (3DGC).”

However, attaching or detaching a window to or from a graphics context is very different from “detaching the custom glass pane from a previous window of the Java application; and attaching the custom glass pane to a next window of the Java application where the mouse cursor is currently located,” as recited in claims 4 and 18 (emphasis added) and similar recitation in claim 12. Doyle does not teach or suggest the features as implemented in a Java application. Withdrawal of the rejection of claims 4, 12, and 18 under 35 U.S.C. §103 (a) is respectfully requested.

Claims 8, 14, and 20 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Schumacher in view of Gallo and Microsoft Windows NT, and further in view of U.S. Patent 6,151,021 to Berquist et al. (hereafter Berquist). This rejection is respectfully traversed.

Berquist is directed to a method for implementing indexing and topography of software notes. Similar to Gallo and Microsoft Windows NT, Berquist cannot be applied to a Java application. Therefore, Berquist does not cure the defect of Schumacher, Gallo, and Microsoft Windows NT and does not disclose or suggest all of the features of amended claims 1, 9, and 15. Claims 8, 14, and 20 are allowable because they depend from allowable claims 1, 9, and 15, respectively, and for the additional features they recite. Withdrawal of the rejection of claims 8, 14, and 20 under 35 U.S.C. §103 (a) is respectfully requested.

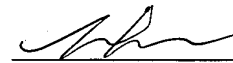
New claims 21-22 and 23-24 are allowable because they depend from allowable claims 1 and 9, respectively, and for the additional features they recite. For example, none of the cited references teach or suggest “wherein the drag image is made half-transparent by changing alpha channel values for each pixel of an original image,” as recited in new claims 21 and 23.

In view of the above remarks, Applicant respectfully submits that the application is in condition for allowance. Prompt examination and allowance are respectfully requested.

Should the Examiner believe that anything further is desired in order to place the application in even better condition for allowance, the Examiner is invited to contact Applicant's undersigned representative at the telephone number listed below.

Respectfully submitted,

Date: **October 18, 2004**



Kelly T. Lee
Registration No. 47,743
Andrews Kurth LLP
1701 Pennsylvania Ave, N.W.
Suite 300
Washington, DC 20006
Tel. (202) 662-2736
Fax (202) 662-2739